Computing, Telecommunication and Control, 2025, Vol. 18, No. 3, Pp. 36–45.
Информатика, телекоммуникации и управление. 2025. Том 18, № 3. С. 36–45.

Research article
DOI: https://doi.org/10.18721/JCSTCS.18303
UDC 004.932

AUTOMATION OF BIOLOGICAL CELL IMAGE PROCESSING

K.A. Turchinskii ✉ , *A.Ye. Krasnov*

Russian State Social University (RSSU), Moscow, Russian Federation

✉ turchin.sky@yandex.ru

Abstract. When analyzing images of biological cells, automated methods for segmentation and result storage are becoming increasingly in demand. Manual annotation is extremely labor-intensive and does not scale to large volumes of data, while conventional segmentation algorithms create binary masks of substantial size. The objective of this work is to develop a software pipeline that combines local threshold filtering and morphological post-processing to obtain an accurate binary mask and then encodes the result using run-length encoding (RLE) to reduce storage space. Methods used are as follows: at the segmentation stage, local statistical criteria are applied, followed by morphological closing. For storing the result, several modifications of RLE (standard, Foreground-Only, DRLE and Z-order) are implemented along with their comparative analysis. The scientific novelty of the work lies in the comprehensive integration of block filtering and morphology with subsequent compression of binary segmentation masks in the task of erythrocyte (and other cells) segmentation. This approach significantly reduces storage requirements without substantial loss of accuracy. The proposed solution demonstrates high metrics (Accuracy, IoU, Dice) while substantial memory savings. The practical significance is that the developed software pipeline can be easily integrated into biomedical data analysis systems, accelerating the mass processing of cell images and reducing the demands on storage infrastructure.

Keywords: segmentation, biological images, run-length encoding, local threshold filtering, morphological post-processing, automation, accuracy

Citation: Turchinskii K.A., Krasnov A.Ye. Automation of biological cell image processing. Computing, Telecommunications and Control, 2025, Vol. 18, No. 3, Pp. 36–45. DOI: 10.18721/JCSTCS.18303

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.18303
УДК 004.932

АВТОМАТИЗАЦИЯ ОБРАБОТКИ ИЗОБРАЖЕНИЙ БИОЛОГИЧЕСКИХ КЛЕТОК

К.А. Турчинский ✉ , *А.Е. Краснов*

Российский государственный социальный университет (РГСУ),
Москва, Российская Федерация

✉ turchin.sky@yandex.ru

Аннотация. При анализе изображений биологических клеток все более востребованными становятся автоматизированные методы сегментации и хранения результатов. Ручная разметка чрезвычайно трудоемка и не масштабируется на большие объемы данных, а обычные алгоритмы сегментации создают бинарные маски значительного объема. Целью работы является разработка программного конвейера, который сочетает локальную пороговую фильтрацию и морфологическую постобработку для получения точной бинарной маски, а затем кодирует результат отрезками разной длины (RLE) для уменьшения занимаемого пространства. Используемые методы: на этапе сегментации применяются локальные статистические критерии, за которыми следует морфологическое закрытие. Для хранения результата внедряются несколько модификаций RLE (стандартная, Foreground-Only, DRLE и Z-order) с их сравнительным анализом. Научная новизна работы заключается в комплексном объединении блоковой фильтрации и морфологии с последующим сжатием бинарных сегментационных масок в задаче сегментации клеток, что позволяет существенно сократить объем хранения без значимого ущерба точности. Полученное решение демонстрирует высокие метрики (Accuracy, IoU, Dice) при существенной экономии памяти. Практическая значимость исследования состоит в том, что разработанный программный конвейер легко интегрируется в системы анализа биомедицинских данных, ускоряя массовую обработку изображений клеток и снижая требования к инфраструктуре хранения.

Ключевые слова: сегментация клеток, биологические изображения, кодирование отрезками разной длины, локальная пороговая фильтрация, морфологическая постобработка, автоматизация, точность

Для цитирования: Turchinskii K.A., Krasnov A.Ye. Automation of biological cell image processing // Computing, Telecommunications and Control. 2025. Т. 18, № 3. С. 36–45. DOI: 10.18721/JCSTCS.18303

Introduction

Digital processing of cell images is becoming increasingly important in biological and medical practice: from visualization of erythrocytes and leukocytes to analysis of tissues and micro-objects [1, 2]. Manual image annotation is an extremely labor-intensive process, which becomes more complicated as data volumes grow. Therefore, there is a demand for automated pipelines that can not only segment target regions (e.g., cell boundaries), but also efficiently store the results as binary masks.

However, traditional approaches to obtaining and storing masks face the problem of data redundancy: even single objects in large fields of view can lead to significant memory costs [3, 4]. To solve this problem, run-length encoding (RLE) and its various modifications – Foreground-Only, DRLE, Z-order [2, 3, 4, 5] – are widely used. At the same time, not only the RLE method itself is important, but also the quality of pre-segmentation, which directly affects the structure of the binary mask and potential compression efficiency.

Medical and biological image processing imposes additional accuracy requirements: incorrectly defined cell boundaries lead to errors in subsequent analyses (e.g., in cell concentration calculation, cell shape estimation etc.) [2]. Consequently, in order to develop reliable automated processing systems, it is necessary to combine high segmentation accuracy with optimized storage. Methods such as local threshold filtering and morphological post-processing, which have proved their effectiveness when working with heterogeneous images[1], can improve segmentation results and prepare them for more efficient compression.

The relevance of the topic under study is driven by the need to improve performance and reduce memory costs in image processing systems that require long-term storage of a large number of segmented cell images. The development of this approach facilitates the scaling of laboratory and clinical studies, enable rapid processing of microscopic observations and support the creation of databases for subsequent analysis, diagnosis and training of neural network models [4]. This creates a practical demand for a comprehensive solution: automatic cell segmentation followed by efficient mask encoding and compressed storage.

The aim of this work is to develop a software pipeline that combines local threshold filtering and morphological post-processing to obtain an accurate binary mask, and then encodes the result using RLE to reduce storage space.

To achieve this goal, the following tasks were solved:

1. To design an algorithm that combines adaptive segmentation and morphological operations with further RLE.

2. To create a prototype of a software package including filtering, contour extraction, binary mask packing and quality metrics computation functions.

3. To test the pipeline on real biological images and evaluate the achieved compression and the compliance of the segmented regions to the reference data.

4. To analyze the influence of filtering and morphology parameters on the error magnitude and volume of encoded masks in order to formulate practical recommendations.

Description of the proposed algorithm

A system processing a discrete image $Im_{m,n}$ of size $M \times N$ pixels (where $m = 1, 2, ..., M$ is the row index and $n = 1, 2, ..., N$ is the column index) will face the problem of brightness inhomogeneities (e.g., uneven illumination and noise). To solve this problem, we introduce a local threshold filtering stage.

Block partitioning is performed as follows. A field of size $M \times N$ is divided into non-overlapping fragments (blocks) of square $B \times B$ or rectangular shape $B_h \times B_w$. In this case, the indices of pixels belonging to the block with number (i, j), approximately satisfy the following expression:

$$i \in \left[(i-1) \cdot B_h + 1, i \cdot B_h\right], \quad j \in \left[(j-1) \cdot B_w + 1, j \cdot B_w\right], \tag{1}$$

where B_h and B_w are the block height and width, and the total number of blocks vertically is equal to M/B_h, horizontally – to N/B_w.

Fig. 1 shows the partitioning of the original image into blocks of size $B \times B$ with a "step" of B, meaning adjacent blocks follow one another without overlap. For each block $\{i, j\}$, the average brightness $\overline{\mathrm{Im}}_{i,j}$ and local standard deviation $\sigma_{i,j}$ are calculated. Next, the local threshold $\theta_{i,j}$ is formed using the following formula:

$$\theta_{i,j} = \overline{\mathrm{Im}}_{i,j} - \alpha\sigma_{i,j}, \tag{2}$$

[1] Voxel Compression, Available: https://eisenwave.github.io/voxel-compression-docs/rle/space_filling_curves.html (Accessed 19.09.2025).

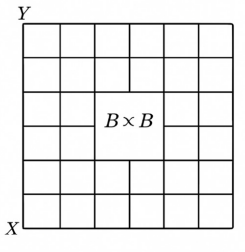

Fig. 1. Schematic partitioning of the original image of size $M \times N$ into non-overlapping blocks $B \times B$ (step = B)

where α is the coefficient adjusting noise sensitivity. Similar expressions are used in a number of adaptive thresholding methods of object extraction[2] [5].

Each pixel (m_i, n_i) inside the current block $\{i, j\}$ is compared with the threshold $\theta_{i,j}$ (3). If

$$\mathrm{Im}_{m_i,n_i} \le \theta_{i,j}, \tag{3}$$

then the pixel is considered to belong to the "cell" region; otherwise, it is assigned to the "background". As a result of this step, an intermediate binary map A is generated, where each pixel is assigned a value of 1 (object) or 0 (background).

The output of the local threshold filtering is a binary map $A(M \times N)$ (4), where

$$A_{m,n} = \begin{cases} 1 : \mathrm{Im}_{m,n} \le \theta_{i,j} \\ 0 : \mathrm{Im}_{m,n} > \theta_{i,j} \end{cases}. \tag{4}$$

Subsequently, morphological processing is used to clean A and eliminate "gaps" or "outliers".

Morphological closing (Close) is usually applied when it is necessary to "connect" adjacent white regions (objects). In its classical form, the closing operation is defined by the following formula:

$$\mathrm{Close}(A) = (A \oplus S) \ominus S, \tag{5}$$

where A is the original binary map (see above); S is a structuring element, such as a circle (disk) or ellipse; $\oplus$ is morphological dilatation (expansion); $\ominus$ is morphological erosion (reduction).

Visually, closing "fills in" small gaps inside an object, thereby merging segments.

Morphological opening $\left(\mathrm{Open}(A) = (A \ominus S) \oplus S\right)$ is used to "clean" the map from random small noises (outliers) that have a small area and do not belong to real cells [6].

The choice between closing, opening or their combination is determined by the characteristic sizes of the cells and the noise content. If it is known that cells can be closely packed, applying closing is advisable. If it is necessary to get rid of "point-like" fragments, adding opening is useful. The final contour of each region, and therefore the structure of the final binary mask, directly depends on the morphology settings.

When the morphologically processed binary map A (the "final mask") is ready, it is subjected to packing using RLE method.

Before encoding, a pixel traversal scheme is defined, for example:

1. Row-major – left to right, top to bottom (by rows).

[2] Voxel Compression, Available: https://eisenwave.github.io/voxel-compression-docs/rle/space_filling_curves.html (Accessed 19.09.2025).

2. Z-order (Morton-order) – pixels are rearranged into a special sequence, more favorable for certain object configurations [3, 8].

During RLE, each consecutive pixel fragment with the same value (0 or 1) is replaced with a "run". In its classical form, this is a pair (r, v), where r is the number of consecutive identical bits, and v is the bit itself (0 or 1). In Foreground-Only encoding [4], where only the "runs" of units are stored, records of the form (s, l) are used, where s is the start index of a block of unit pixels, and l is the length of this continuous block. A variant of differential RLE additionally encodes the difference between the lengths of consecutive runs, which sometimes reduces the overall data volume [6]. Z-order encoding is often useful if objects are highly "clustered" and arranged in compact groups.

The criterion for choosing the encoding method (DRLE, Z-order, Foreground-Only etc.) depends on the number of cells, their sizes, noise density and decoding frequency. If decoding needs to be performed frequently, then the simplest format RLE(r,v), which does not require complex permutations or calculations, is more convenient.

The final stage is packing and saving the run-length data (i.e., a set of pairs like (r, v) or (s, l) etc.) in a form convenient for storage. In applied scenarios, this may be:

– structured record;
– serialization;
– general "stream" compression.

The essence of "run-length data" is a list (or other structure) of segments with the same bit. Because there are often extensive areas of background outside the cells, RLE achieves significant memory savings. When there are many disparate tiny objects, other options (e.g., Foreground-Only) are chosen to avoid storing long chains of zeros.

The modular architecture of the developed solution provides flexibility:

1. Local threshold filtering is easily adjustable to cell size and contrast.
2. Morphological operations can be varied (use opening, closing or their combination, changing the type of structuring element).
3. RLE system can be replaced or supplemented with other compression methods (e.g., Quadtree, Octree or ZIP archiving).

This simplifies the adaptation of the algorithm to different types of biological images, as well as scaling for large data volumes. If other tools (non-Python) have to be used in the future, it is still convenient to have run-length data, as it is easy to import and unpack in most environments.

It is convenient to describe the technological pipeline using a diagram reflecting the sequence of blocks and principles of interaction (Fig. 2).

The loading module accepts a list of source files stored in any available graphic formats and sends each image sequentially to the local threshold filtering stage. A block-based or adaptive strategy is used to flexibly adjust the brightness threshold when different frame zones have uneven intensity. The morphological module eliminates minor defects that may arise from random noise or optical system artefacts. Closing with a suitable structuring element allows merging adjacent objects to form more solid cell regions, while opening filters out excess details protruding beyond the intended contours.

The resulting system is capable to operate in batch mode, sequentially processing an extensive collection of images and generating grouped results in a convenient format for analysis or further transfer. The practical significance lies in resource savings: storage capacity, transfer speed and processing requirements become more manageable, while the preservation of contours allows for statistics on cell number and shape to be conducted.

Architecture and implementation of the software package

The foundation of the work is implemented through a set of modules, each performing its own functions for segmentation, packing the results and interacting with the user interface. One of the key

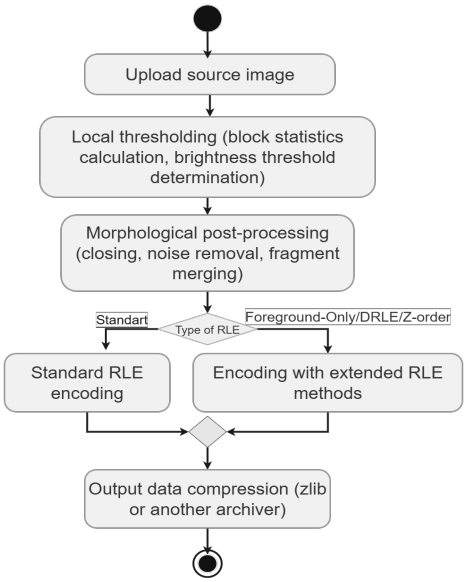

Fig. 2. Algorithm for automation of biological image processing

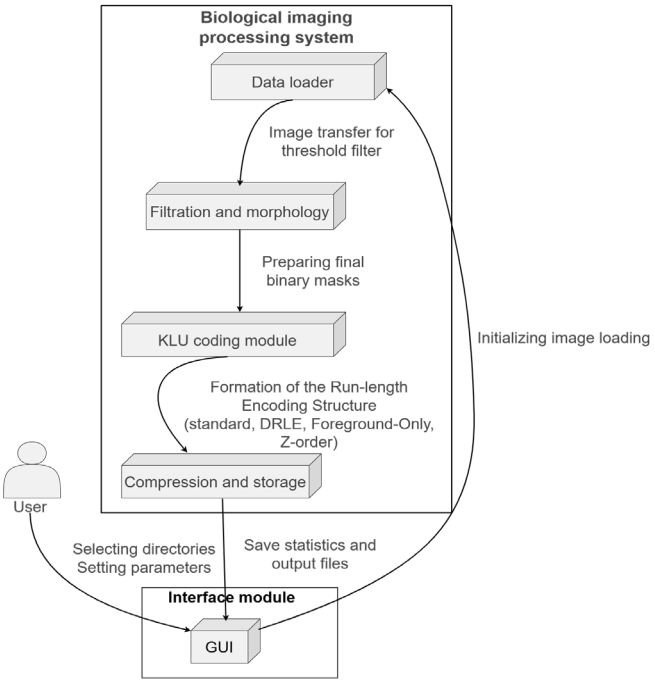

Fig. 3. Architecture of the software package

requirements is that the listed stages, including morphological correction and various RLE schemes, are flexible in configuration and can be extended with additional procedures if necessary. The architecture diagram in Fig. 3 reflects the overall flow of transformations.

The calling process, initiated by user actions, involves reading the necessary files in the "Data Loader" block. The local or remote directory access module is configured to read images in batch

mode, and all intermediate results and service metadata are collected into a single structure. Next, the data enters the node responsible for threshold filtering and morphological operations. It is at this stage that local brightness estimation, adaptive threshold calculation, closing of detected gaps and removal of noise inclusions occur, if they do not match the expected cell contours.

Encapsulation of different RLE schemes within a separate module simplifies further maintenance, as it is easy to add new packing variants or special optimization modes to the working pipeline, depending on the features of the images under study. All types of RLE conclude with the formation of a structure, which is packed using zlib or other compression methods. The "Compression and Storage" module outputs the result in a form convenient for storage and transfer, while informational responces, e.g., statistics on the number of processed frames, final packet size, conversion time, are sent back to the "Interface Module".

The software implementation relies on image processing libraries (OpenCV, Pillow), as well as tools for mathematical calculations (NumPy) and serialization (pickle). Such a suite ensures operation both on local machines and on server systems, where it is necessary to scale the processing of large sets of microphotographs. At the same time, the same encoding module can work on data that has undergone different types of morphological correction. This method is suitable for situations where different cell types have a particular contour structure and require the selection of specific morphology or filtering settings.

Experimental results and analyses

The system pipeline was tested on a dataset consisting of 343 annotated biological images (BloodImage_XXXX). For each image, stored alongside the corresponding XML annotation, a binary mask reflecting the location of cells (RBCs) in the frame was automatically generated. Local threshold filtering with fixed block size and morphological closing to correct small gaps within regions were the key steps. Subsequently, the resulting maps were subjected sequentially to classical RLE, Foreground-Only, DRLE, Z-order RLE and a combined method incorporating local threshold filtering with RLE. It was important to capture computational cost, compressed file size and accuracy metrics to understand the relationship between packing efficiency and cell structure preservation. Table 1 shows the main results.

Table 1

Comparative analysis with the proposed method

Parameters	Standard RLE	Foreground-Only RLE	DRLE	Z-order RLE	RLE Enhanced Local Filtering + RLE
Total encoding time, seconds	9.4137	7.7363	9.3730	9.8173	10.2493
Total decode time, seconds	0.3608	0.1910	0.4326	0.9448	0.3835
Average size, bytes	406.36	2791.14	512.78	1525.84	391.92
Average accuracy, %	100.00	100.00	100.00	100.00	99.54
Average *IoU*	1.00	1.00	1.00	1.00	0.9873
Average *Dice*	1.00	1.00	1.00	1.00	0.9936

The full sample included 343 XML descriptions. The processing time recorded by the system took approximately one minute at an average speed of about 6 images per second. Sequential encoding-decoding stages were considered, and the average size in bytes was calculated to get an idea of typical compressed mask volumes. Improved local threshold filtering in the last method (the fifth variant) additionally allowed for the estimation of accuracy (acc), IoU and Dice, as its own procedure modified the original boundaries.

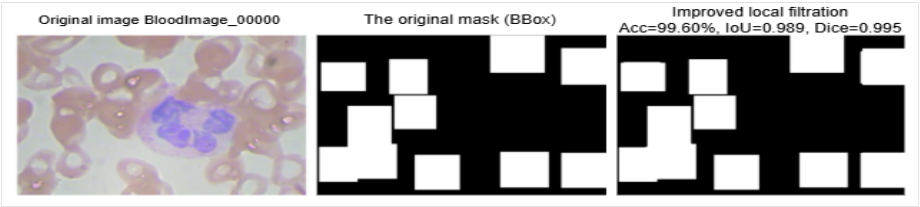

Fig. 4. Visualization of image processing by the proposed method

Combined analysis of encoding costs and file size confirmed that classical RLE yielded a fairly modest volume indicator (406.36 bytes on average) with a noticeably lower decoding time (0.3608 sec). However, the Foreground-Only scheme presented a failure: the average size increased to 2791.14 bytes, indicating a high number of disparate fragments requiring separate description. DRLE maintained a moderate compactness (512.78 bytes) with a slight increase in decoding time (0.4326 sec). Z-order increased the size to 1525.84 bytes, most likely because mixing local mask blocks did not always lead to longer runs in the overall structure. The standard schemes (Standard RLE, Foreground-Only RLE, DRLE and Z-order RLE) are lossless: they do not alter the binary mask and hence exhibit maximum accuracy (acc = 100%, IoU = 1.00, Dice = 1.00). Nevertheless, their average size varies from 406 to 2791 bytes. The proposed "local threshold filtering + morphology + RLE" complex yields the smallest average size of 391.92 bytes with near-maximum accuracy (acc = 99.54%, IoU = 0.9873, Dice = = 0.9936), which we consider as an optimal compromise between memory savings and the preservation of diagnostically significant contours. For each scheme, the accuracy metrics are equal to one because the encoding-decoding operation does not change the original partitioning. We have included these metrics in Table 1 to ensure a correct comparison by size + accuracy criteria.

Contour overlap (Dice = 0.9936) was particularly impressive, indicating that on average the final mask matched the reference data almost without distortion. Hence, local filtering and morphology not only preserved the integrity of cell regions, but also contributed to reducing the volume of the recorded mask due to larger homogeneous areas. The assumption that closing eliminates internal gaps and thus increases run lengths was confirmed empirically. The data obtained for Z-order indicated that Morton permutation is not always optimal, especially when cells are located in different regions and their mutual geometry does not create large connected clusters.

A visual comparison of the segmentation is presented in Fig. 4, where the original frame contains a blood fragment with individual cellular elements, and the resulting masks demonstrate the importance of merging closely located areas to improve boundary consistency.

The above example confirms the conclusions drawn from the encoding time and size statistics. The improved local filtering that corrects illumination variations within the image eliminates minor noise objects, resulting in a clearer binary map and longer sequences of 1's or 0's during subsequent RLE.

The results indicate the high efficiency of the proposed integrated approach, where filtering and morphology prepare the image for RLE. The obtained metrics facilitate the large-scale automation of cell analysis with big data volumes and help save memory or transmission channels without the risk of losing key diagnostic details. This strategy is particularly appropriate in the biomedical field, where precise contours are required while the compactness of the recorded structures is also important.

Conclusions

The testing of the complex solution based on local threshold filtering, morphological post-processing and subsequent mask encoding using different RLE schemes demonstrates a significant gain when

working with large arrays of microscopic images. Statistical data for 343 annotated images confirm that careful noise removal within segmented regions and correct fragment smoothing have a noticeable effect on reducing the final volume, while improving accuracy metrics and preserving cell configuration. Applying classical RLE to masks that have undergone additional local thresholding and morphological closing procedures results in a relatively small file size and an almost complete match with the reference boundaries.

The observed advantage over alternative methods is explained by the fact that local filtering adjusts to the details of each image fragment, eliminating global thresholding errors, while closing merges disparate pixels into large regions where the runs become longer. Morton (Z-order) permutation can be convenient for strictly localized clusters, but in some cases does not provide improved compression. Foreground-Only scheme is effective when the background occupies almost the entire area and cells are concentrated in one region, but leads to redundant intervals for fragmented objects. The results confirm the need for an integrated approach: each module (filtering, morphology, RLE) enhances the positive effect of the other elements. The studied pipeline simplifies the large-scale automation of medical analysis tasks and opens up opportunities for integration into laboratory and clinical systems oriented to the rapid processing and storage of a large number of biological images.

REFERENCES

1. **Haggere L.R., Alagarswamy R.** HWCD: A hybrid approach for image compression using wavelet, encryption using confusion, and decryption using diffusion scheme. *Journal of Intelligent Systems*, 2023, Vol. 32, No. 1, Art. no. 20229056. DOI: 10.1515/jisys-2022-9056

2. **Rahman M.** Comparative analysis of run-length encoding techniques: Standard, modified, two-dimensional, and bitwise approaches for efficient data compression with numerical example, 2025.

3. **Thong Y.J.** An introduction to bitmask representations and encodings: RLE vs REE. Datature, 2024. Available: https://www.datature.io/blog/an-introduction-to-bitmask-representations-and-encodings-rle-vs-ree (Accessed 19.09.2025).

4. **Schmidt K., Horowitz J.** *Method and apparatus for double run-length encoding of binary data*. European Patent, No. EP0783208A2, 1997.

5. **Haghighi K.G., Mirnia M.K., Navin A.H.** Optimizing run-length algorithm using octonary repetition tree. *arXiv:1611.09664*, 2016. DOI: 10.48550/arXiv.1611.09664

6. **Samson A.Ch., Sastry V.U.K.** An improved run length encoding scheme for image compression. *International Journal of Engineering and Computer Science*, 2017, Vol. 6, No. 3, Art. no. 57. DOI: 10.18535/ijecs/v6i3.57

7. **York T.** Quadtrees for image processing. *Medium*, 2020. Available: https://medium.com/@tannerwyork/quadtrees-for-image-processing-302536c95c00 (Accessed 19.09.2025).

8. **Shen J.** Run-length encode and decode. *Medium*, 2021. Available at: https://medium.com/@ccshenyltw/run-length-encode-and-decode-a33383142e6b (Accessed 19.09.2025).

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Kirill A. Turchinskii
Турчинский Кирилл Александрович
E-mail: turchin.sky@yandex.ru

Andrey Ye. Krasnov
Краснов Андрей Евгеньевич
E-mail: krasnovmgutu@yandex.ru

Submitted: 21.04.2025; Approved: 24.07.2025; Accepted: 18.09.2025.
Поступила: 21.04.2025; Одобрена: 24.07.2025; Принята: 18.09.2025.